

application being unshared and without routing by the terminal server of input/output for the second application to the clients of the terminal server;

9. cont  
determining, by the second application, a session identifier for a client sharing the first application within the terminal server environment; and

using the session identifier to send a message to the client instead of to the terminal server.

2. (Unchanged) The method of claim 1, wherein the first application is unaware it is being shared.

3. (Amended) The method of claim 1, further comprising:  
using the session identifier to establish an input/output communication channel with the client.

4. (Unchanged) The method of claim 3, further comprising:  
receiving over said communication channel a response to the message.

5. (Unchanged) The method of claim 3, further comprising:  
monitoring, by the second application, of accessing of resources by the first application; and  
determining, by the second application, an error condition arising from accessing a particular resource by the first application;  
wherein the message concerns the error condition and the message is sent to the client over said communication channel.

6. (Unchanged) The method of claim 5, wherein the second application is a virus scanner, and wherein the error condition is a virus detected in the particular resource.

7. (Unchanged) The method of claim 1, further comprising:

monitoring, by the second application, of accessing of resources by the first application; and

determining, by the second application, an error condition arising from accessing a particular resource by the first application;

wherein the message concerns the error condition.

8. (Unchanged) The method of claim 7, wherein the second application is a virus scanner, and wherein the error condition is a virus detected in the particular resource.

9. (Unchanged) The method of claim 7, further comprising:  
starting an elapsed-time counter; and  
starting scanning the particular resource for viruses;  
wherein said determining the error condition comprises identifying the elapsed-time counter has exceeded a scanning time-limit.

10. (Unchanged) The method of claim 7, further comprising:  
starting scanning the particular resource for viruses; and  
determining if the particular resource corresponds to an archive file, and if so, starting an elapsed-time counter before scanning the archive file for viruses;  
wherein said determining the error condition includes determining if the elapsed-time counter exceeded a scanning time-limit.

a2  
11. (Amended) A readable medium having encoded thereon instructions for allowing unshared applications executing within a terminal server environment to interact with clients of a terminal server, wherein an application shared within the terminal server environment is executed as multiple instances on the terminal server instead of on the clients and the terminal server routes input/output for each instance to the client associated with the instance, said instructions when executed capable of directing a processor to:

execute a second application on the terminal server to interact with a first application shared within the terminal server environment, said second application being unshared and without routing by the terminal server of input/output for the second application to the clients of the terminal server;

determine, by the second application, a session identifier for client sharing the first application; and

use the session identifier to send a message to the client instead of to the terminal server.

12. (Unchanged) The medium of claim 1, said instructions comprising further instructions to direct the processor to:

use the session identifier to establish an input/output communication channel with the client.

13. (Unchanged) The medium of claim 12, said instructions comprising further instructions to direct the processor to:

receive, over said communication channel, a response to the message.

14. (Unchanged) The medium of claim 12, said instructions comprising further instructions to direct the processor to:

monitor, by the second application, accessing of resources by the first application;

determine, by the second application, an error condition arising from accessing a particular resource by the first application;

configure the message to include the error condition; and

send the message over said communication channel.

15. (Unchanged) The medium of claim 14, wherein the second application is a virus scanner, and wherein the error condition is a virus detected in the particular resource.

16. (Unchanged) The medium of claim 11, said instructions comprising further instructions to direct the processor to:

- monitor, by the second application, of accessing of resources by the first application; and
- configure the message to include the error condition; and
- send the message over said communication channel.

17. (Unchanged) The medium of claim 16, wherein the second application is a virus scanner, and wherein the error condition is a virus detected in the particular resource.

18. (Unchanged) The medium of claim 16, said instructions comprising further instructions to direct the processor to:

- start an elapsed-time counter; and
- start scanning the particular resource for viruses;

wherein said instructions for determining the error condition further comprise instructions for determining that the elapsed-time counter has exceeded a scanning time-limit.

19. (Unchanged) The medium of claim 16, said instructions comprising further instructions to direct the processor to:

- start scanning the particular resource for viruses; and
- determine if the particular resource corresponds to an archive file, and if so, starting an elapsed-time counter before scanning the archive file for viruses;

wherein said instructions for determining the error condition further comprise instructions for determining that the elapsed-time counter has exceeded a scanning time-limit.

20. (Amended) A system for unshared applications executing within a terminal server environment to interact with clients of a terminal server, wherein an application shared within the terminal server environment is executed as multiple

instances on the terminal server instead of on the clients and the terminal server routes input/output for each instance to the client associated with the instance, the system comprising:

a file access monitor for monitoring file accesses by a first application shared within the terminal server environment;

a virus scanning arrangement executing on the server for scanning accessed files for viruses;

a timer arrangement for timing said scanning accessed files for viruses;

a scan-termination arrangement for interrupting the virus scanning arrangement if said scanning accessed files for viruses does not complete within a timeout period;

means for determining, by the virus scanning arrangement, a session identifier for a client sharing the first application; and

means for sending a message, to the client according to the session identifier instead of to the terminal server, indicating said scanning accessed files for viruses timed out.